Multi-destination secure electronic mail

C. MITCHELL
Hewlett-Pockord Laboratories, Filton Road, Stoke Gifford, Bristol BS12 6QZ

Electronic mail messages are often sent to more than one destination; this gives rise to problems when security is required. A recent draft for a standard for securing electronic mail messages suggests a novel mechanism for solving the problem. Unfortunately, as shown herein, the solution is flawed and can allow the construction of fake messages which will pass the authenticity tests.

Received May 1987, revised July 1987

1. INTRODUCTION

In most if not all of today's electronic mail systems there exists the capability of sending a message to a list of users simultaneously. When such a message is sent, it will often only be replicated when it really needs to be, so that a single message sent from the U.K. to two recipients in the U.S.A. will only be made into two copies after it has crossed the Atlantic.

Such a process is obviously desirable, not least because of the saving in information that needs to be sent across potentially heavily loaded communications links. However, a problem arises when a message containing sensitive information needs to be encrypted and/or authenticated to protect it against disclosure and/or alteration whilst in transit.

In most envisaged secure communication systems a pair of users who wish to communicate securely are equipped with a key known only to them. This key can then be used with an encryption algorithm to encrypt and/or authenticate messages.

So if a message is to be sent securely to two different users, then it will need to be encrypted using two different keys. Perhaps the most obvious solution to this problem is to replicate the message locally, i.e. to send two copies of the message encrypted appropriately. This has a considerable overhead for local communications, particularly for messages sent to large groups of users.

Another relatively simple solution is to require the provision of keys for groups of users who commonly communicate with one another. Thus, when a message is to be sent to a particular group of users, the key specific to that group is used in the encryption process, and the message only needs to be sent once.

Unfortunately, a number of problems arise with this solution. First, and perhaps most importantly, consider the situation when it is necessary for a user to be removed from a set of mailing lists. This will require the redistribution of all the group keys owned by that user. This could be both very costly and difficult to manage. The second problem is that the number of groups involved may grow very large. In many situations almost every message is sent to a slightly different list of people, and the number of group keys required could very easily become prohibitive.

It is clear that some other, simpler solution to the multi-destination problem is required. We consider below one ingenious solution proposed in a recent Request For Comments (RFC) for the DARPA 'Internet' electronic mail system.[1] However, as we shall see, this solution has a flaw which in certain circumstances allows the malicious construction of apparently authentic messages.

Finally note that it is not the intention of the author to criticise what is after all only an RFC,[1] but rather to ensure that possible solutions to multi-destination secure message authentication problems are analysed with care. There are many lessons to be learnt from the study of flawed systems which are apparently perfectly secure.

2. A PROPOSED SOLUTION AND ITS WEAKNESS

We describe here the essential details of the system described in the draft standard,[1] although many of the statements made below are of more general application. Interestingly, the basic idea for solving the multi-destination secure mail problem has been independently invented at least twice, although it does not appear in any literature known to the author.

We first suppose that every pair of users, A, B say, wishing to exchange secure mail are equipped with a secret *Interchange Key* (IK), which we denote by KAB. It is not important how this key is distributed, but we assume that each key is known only to the appropriate pair of users (and perhaps to a Key Distribution Centre, if one exists). All messages are to be encrypted and/or authenticated using a block cipher in *Cipher Block Chaining* (CBC) mode, where the block cipher algorithm to be used is immaterial, but it could, for example, be the DES algorithm.[4,5] Briefly, use of CBC means that the message is first divided into a sequence of n-bit blocks, $M_1, M_2, \ldots, M_r$, say, where n is the cipher block length (e.g. for DES, $n = 64$). To perform the encryption or authentication operation requires a key for the cipher (K say) and an n-bit *Initialisation Vector* (IV), which for convenience we call C_0. Note that this IV is either pre-agreed or sent with the message in an unencrypted form. If CBC is to be used for encryption then the ciphertext is $C_1, C_2, \ldots, C_r$ where $C_i = \{M_i\}^K + C_{i-1}$, '+' denotes bit-wise X-or, and, as throughout this paper, $\{M\}^K$ denotes the effect of enciphering block M using key K. If CBC is to be used for authentication, then the *Message Authentication Code* (MAC) is equal to all or part of C_r. For further details on CBC see, for example, the standard modes of use for the DES algorithm.[2,3]

When a message is sent from user A to user B in a secure fashion, the following procedure is followed. Note

that there are basically two options for message security: either (i) authentication with no encryption, or (ii) both authentication and encryption; i.e. all secure messages are authenticated. We first consider option (i).

1. A random *Data Encrypting Key* (DEK) is obtained by A which is to be used to secure this message (and no other message).
2. This DEK is encrypted using the block cipher in Electronic Codebook Mode,[2,4] under KAB.
3. The message is encrypted using the block cipher in CBC mode under the DEK with Initialisation Vector (IV) set to all zeros. All the ciphertext blocks except the last one are discarded, with the remaining block forming the *Message Authentication Code* (MAC), which is used to authenticate the message to the receiver; this process is well established.[6] Finally, the MAC is encrypted using the block cipher in ECB mode under the control of KAB.
4. As encryption of the message is not required it is now sent in clear form, preceded by the DEK encrypted as in (2) and the MAC computed and encrypted as in (3). The transmitted data will therefore have the form:

$$\{\text{DEK}\}^{\text{KAB}}, \{\text{MAC}\}^{\text{KAB}}, \text{message}.$$

If option (ii) is required, i.e. if encryption is also required, then steps (1), (2), (3) above are performed, followed by steps (5), (6), (7) given below.

5. The DEK is modified by inverting the bits in alternate nibbles to obtain the *Modified DEK* (MDEK).
6. The clear message is encrypted again using CBC, but this time under control of the MDEK (rather than the DEK) together with a randomly chosen IV.
7. The message is now sent, encrypted as in (6), and preceded by the DEK encrypted as in (2), the MAC computed and encrypted as in (3) and the randomly chosen IV used in (6). The transmitted data will therefore have the form:

$$\{\text{DEK}\}^{\text{KAB}}, \{\text{MAC}\}^{\text{KAB}}, \text{IV}, \{\text{message}\}^{\text{MDEK}}.$$

This completes the security process for messages sent from one user to another single user. We now consider how the process is modified in the case that user A wishes to send the same message to users B and C (the process works equally well for more than two users, and is straightforward to generalise to that case).

For authentication only, step (1) as above is performed followed by steps (2*), (3*) and (4*) given below.

2*. Two encrypted versions of the DEK are produced using the block cipher in Electronic Codebook Mode,[2,4] once under KAB and once under KAC.
3*. The message is encrypted using the block cipher in CBC mode under the DEK with Initialisation Vector (IV) set to all zeros. All the ciphertext blocks except the last one are discarded, with the remaining block forming the MAC which is used to authenticate the message to the receiver. Finally, two encrypted versions of the MAC are produced using the block cipher in ECB mode, once under KAB and once under KAC.
4*. As encryption of the message is not required it is now sent in clear form, preceded by the two encrypted versions of the DEK (computed as in (2*)) and the two encrypted versions of the MAC (calculated as in (3*)). The transmitted data will therefore have the form:

$$\{\text{DEK}\}^{\text{KAB}}, \{\text{MAC}\}^{\text{KAB}}, \{\text{DEK}\}^{\text{KAC}}, \{\text{MAC}\}^{\text{KAC}}, \text{message}.$$

If option (ii) is required, i.e. if encryption is also required, then steps (1), (2*), (3*) above are performed, followed by steps (5), (6) and (7*), the last of which is given below.

7*. The message is now sent, encrypted as in (6), and preceded by the two encrypted versions of the DEK (see (2*)), the two encrypted versions of the MAC (see (3*)) and the randomly chosen IV used in (6). The transmitted data will therefore have the form:

$$\{\text{DEK}\}^{\text{KAB}}, \{\text{MAC}\}^{\text{KAB}}, \{\text{DEK}\}^{\text{KAC}}, \{\text{MAC}\}^{\text{KAC}}, \text{IV}, \{\text{message}\}^{\text{MDEK}}.$$

Using this modified procedure, each of B and C can use their own IK to recover the DEK used to authenticate (and, if relevant, encrypt) the message. So far so good. However, we now show how user C can use such a message to send a new message to B which B will believe to have come from A. We describe the procedure for authentication only, but the generalisation to encryption should be clear.

C first creates the message which is to be sent to B as if from A. The following procedure is then followed.

1. C recovers the MAC for the original message (this can be done since it was sent encrypted under KAC which is known to C).
2. In a similar way C recovers the DEK used to authenticate the original message.
3. C decrypts the MAC using the DEK to obtain a block we call x.
4. C encrypts the new message using CBC under the control of the DEK to obtain a new MAC we call y.
5. C joins the block $x+y$ on to the end of the new message as an additional 'garbage' block, where the + denotes exclusive or of blocks.
6. C sends to B the new message (augmented by $x+y$) preceded by the DEK encrypted under KAB and the MAC from the original message also encrypted under KAB. These latter pieces of information are present in the original message and hence are known to C.

A message prepared using steps (1)–(6) above will pass B's authentication check using the MAC, and will therefore be accepted as coming from A. The reason for this is straightforward: the CBC encryption of the new message using the DEK will produce final block y, which, when added to the 'garbage block' $x+y$ will give block x which encrypts to the desired MAC.

The one thing suspicious in the message is the 'garbage' block, although even this could be moved to the middle or even the beginning of a message using an extension of the procedure described above. It could well be the case that in most cases messages containing such suspect blocks will be rejected by their recipients, but this does not remove the need to correct the problem. This is because the possible acceptance of such messages, even in

only exceptional cases, means that the system must be regarded as flawed.

It could also be argued that *B* will receive two messages both secured using the same DEK, which would be a rather suspicious event. However, if *C* was in league with the network provider, then *A*'s original message could be prevented from getting to *B*, and no unusual event would be detectable by *B*.

We have therefore discovered a major drawback with the use of the suggested system for sending secure mail to a multiplicity of users. Although it would be desirable to try and 'repair' the above system to preclude the type of fraud described, it is by no means obvious how to do this. The basic problem is that user *C* knows both the MAC and the key used to generate the MAC for the original message, and it is this, in combination with the fact that the CBC function can be inverted, which makes the fraud possible.

Possible secure modifications must either provide distinct authentication keys and MACs for each intended recipient, or use a one-way function to compute the MAC thus preventing the inversion operation. The first possible modification could be very time-consuming since the MAC computation would need to be done for every possible recipient. It is certainly true that all proposed solutions need to be very carefully examined for possible flaws.

Finally, observe that the motivation for one aspect of the system described above is less than obvious, namely the encryption of the MAC under the IK. It is interesting to speculate that this is present to try and prevent the type of fraud described here. Certainly it is true that without this MAC encryption it would be even easier to construct fraudulent messages, since the MAC could be changed without detection and the 'garbage' block would not be necessary.

REFERENCES

1. J. Linn, Privacy enhancement for Internet electronic mail: Part I: Message encipherment and authentication procedures. *Request for comments 989 (RFC 989), IAB Internet Privacy Task Force* (Feb. 1987).
2. ANSI X3.106–1983, *Modes of operation of the DEA.* American National Standards Institute (New York) (1983).
3. FIPS 81, *DES modes of operation.* Federal Information Processing Standard, National Bureau of Standards (Washington, DC) (Dec. 1980).
4. FIPS 46, *Data encryption standard.* Federal Information Processing Standard, National Bureau of Standards (Washington, DC) (Jan. 1977).
5. ANSI X3.92–1981, *Data encryption algorithm.* American National Standards Institute (New York) (1981).
6. ANSI X9.9–1982, *Financial institution message authentication.* American Bankers Association (Washington, DC) (April 1982).

Announcement

12–14 APRIL 1989

ETC 89, the First European Test Conference, Palais des Congrès, Paris, France.

Topics: Component, board and system testing; test development; test systems; design for testability; and new test technologies.

Supported by: SEE, EUREL, IEEE Computer Society, AICA, GME, IEE.

Contact: Colin Maunder, British Telecom Research Labs, Martlesham Heath, Ipswich, Suffolk, IP5 7RE, UK. Tel.: (+44) 473 642706.

12–14 APRIL 1989

Artificial Intelligence and Software Engineering: Promise and Problems. An International Workshop sponsored by the AAAI, University of Exeter.

The purpose of this workshop is to present and discuss a broad set of issues relating to the promise and problems of exploiting AI in practical software. The four foci of the workshop are: AI-based support environments; AI mechanisms and techniques in practical software; software engineering tools and techniques for practial AI software: and methodological issues.

The workshop will be structured around invited presentations from both practitioners and researchers from the USA and from Europe. Each such presentation will be followed by ample discussion time. In addition, some short presentations of relevant submitted papers will be scheduled. Several panel discussions are also planned.

In order to facilitate the possibility of useful, open discussion the workshop will be limited to approximately 40 persons. If you would like to participate, present a paper, or organise a panel discussion, please send a one-page summary of your interests in this area to:

Professor Derek Partridge, Department of Computer Science, University of Exeter, Exeter EX4 4PT, U.K. email: derek@uk.ac.exeter.cs. Tel: 0392 264069; fax: 0392 263108.

12–14 JULY 1989

BNCOD-7, Seventh British National Conference on Databases, Heriot-Watt University, Edinburgh, in association with The British Computer Society.

Papers will be presented on various aspects of databases and database systems. This includes topics such as:

- Deductive databases
- Object-oriented databases
- Multimedia databases
- Knowledge bases
- Expert database systems
- Distributed databases
- Data models
- Database performance
- Information retrieval
- Database design
- Advanced user interfaces
- Geographic/cartographic databases

For further information contact:
Professor M. H. Williams, Computer Science Department, Heriot-Watt University, 79 Grassmarket, Edinburgh EH1 2HJ.